PharmaSUG2010 - Paper MA08

In for the Long Haul:
Retaining a Talented SAS® Programming Team
Lynn DiFinizio, Director, Statistical Programming, Biogen Idec, Cambridge, MA
John Hotaling, Senior Manager, Statistical Programming, Biogen Idec, Cambridge, MA

ABSTRACT
SAS® programmers have a variety of skills, preferences, and communication styles, but there is a universal truth across the demographic – we enjoy programming so much that we stick with it for years, always learning and sharing new and better ways of using SAS®. A team of talented programmers working well together is an extremely valuable asset, one which needs to be developed and protected over time. In this paper, we provide tips for managers/project leaders to build a strong and productive SAS® programming team, retain that programming talent, and develop the team over time. Additionally, we suggest ways to protect your team's ability to continue to work well in the event that one of your key team members wins the lottery and decides to retire from the exciting life of SAS® programming.

INTRODUCTION
Building a talented, effective SAS® programming team is one challenge, but retaining and developing that team over time is an even bigger undertaking. As a manager or project leader, your decisions and actions will be crucial to ensuring that your team is able to work well together to meet objectives, bringing together the right mix of skills to take on the day to day tasks as well as building the “institutional memory” and complex skills needed in any organization to handle ad-hoc requests. If one or more of your most valued and experienced programmers decides to leave your team, this can put goals at serious risk. Therefore, once you have built a highly productive and experienced team, there are things you should be doing to proactively retain your programming talent and to help your team be motivated by their work, as well as to plan for eventual team changes (including succession planning).

The following paper summarizes these tips as six key tenets:

- Plan Ahead
- Communicate Often
- Build “Up and Out”
- Develop Extensively
- Retain and Reward
- Protect the Team

PLAN AHEAD
So you’ve been assigned to lead a team. Congratulations! Before you start ringing up old friends and networking with other programmers to identify who is “good” to be on your team, you should take some time to think through your project, to plan exactly what skills you need and what your programmers individually and as a team need to accomplish together. Not only will this help you focus in on the right skills and individuals but this will also help you to communicate those detailed expectation to your team once you get started.

Overall Team/Goal questions that may help you in detailed planning:

- Why is it important that this team exist? What is the overall end goal? Why is it valuable?
- What are the short term (i.e., 2 weeks, 2 months?) and long term goals (i.e., 2 years)? How achievable are those goals (realistic or “stretch”)?
- How will each person’s work support meeting those goals?
What do you and your team need in order to be successful?
How will you inspire your team individually and as a group?
What will the output of this work be used for? Who or what is the outcome impacting?
What specific programming skills, organizational, writing, etc. skills are needed to do the work?
What will the team need to learn in order to meet the goals?
What groups will the team interact with? What kinds of skills will be needed (i.e., strong communication skills, organizational skills) to work with other departments/team members?
How will you communicate the status/progress of your team?
How will you recognize excellence in your team? Alternatively, how will you improve team members if needed?
How will you escalate issues in your team?
How will you reward the team once they accomplish their goals?
How/when will you need to dissolve the team? Or how will the team need to change over time after completing their goals?

Let’s not forget process and tools specific questions as well:
What process will the team need to follow (or develop)?
Will the team work closely together or more loosely as individuals?
Will each team member own a piece of the work or will they need to collaborate closely?
How will the work be checked for completeness/quality?
What version of SAS® will be used? What SAS® infrastructure (UNIX, PC, etc.)?
What key skills or experiences will the programmers need to have?
How will the team communicate progress or problems?
How much documentation is needed? How will that be easily and globally accessible by the team?
Do team members need to be in the same location or can be global?
What’s the best environment for the team to work (flexible, remote working)?
Overall, how many programmers will be needed to do the work, at what levels and at what locations?

Based on the responses to these and other questions, you can create an extensive, detailed plan that will address both the successful outcome by the team as well as unexpected hurdles, issues, or problems along the way.

COMMUNICATE OFTEN

For programmers to excel, they need to understand what they are being asked to do and why it is important to the project and to the team. If a programmer is engaged and feel their work is valued, he/she will put more effort and care into their work. Also, he/she should understand in detail the expectations of their manager, their project sponsor(s) as well as the project team. The programmer should be held accountable for meeting expectations and communicating issues or problems in a timely fashion.

Programmers tend to function best in an open environment where they are comfortable raising questions, voicing opinions, sharing ideas and requesting or offering help. It will be important as a project leader to provide enough communication on progress, problems, and plans as well as to provide an appropriate forum where programmers can share and solve issues.

Successes should be celebrated. Some programmers thrive on public recognition of their efforts whereas some programmers do not enjoy being in the limelight. Sometimes it is important for even the most reserved programmers
to be recognized publicly so that other members of the team can appreciate what their team members are accomplishing but not bragging about.

The following questions can help you ensure you are communicating sufficiently:

- Have you been communicating regularly about how the team’s work is progressing and the value of that progress?
- Have you confirmed regularly with the team that expectations are clear?
- Do programmers feel comfortable asking questions and raising issues? Do you find out about issues proactively versus “crisis mode”?
- Do programmers in your team share SAS® code and tips to improve their programming?
- Have you communicated examples of excellence to individuals and to the team?
- Have you communicated successes with senior managers/leaders?

BUILD “UP AND OUT”

Programmers like interesting, important, meaningful project work where they can utilize their talents. Rather than have a programmer work on the same types of SAS® programming tasks all of the time because they have excelled at those tasks, you should be planning for programmers to take on various challenges to develop their skills as well as ensure there are backup programmers trained across tasks. Set up an environment/process where programmers are mentoring each other.

When programmers have been doing the same task and working with the same people for a while, they develop efficient habits and resourceful ways of working. However, if this setup leads to complacency (or eventual boredom by a programmer), a manager may want to consider changing the team assignments, reassigning work that a programmer is working on to someone that will view it as a challenge. It will force the team to grow by taking past experiences and applying them to working with new people/new tasks.

Additionally, the application of standard programming techniques by the team can be crucial to making tasks efficient and easily supported by the entire team. Standardized programming techniques facilitate the sharing of programming tasks in case a programmer needs to move on to a higher priority project or a programmer leaves the team. If these standards have been utilized across the programming code developed by the group, other programmers should be able to easily pick up the work.

Lastly, as it is necessary to add new programmers to the team, ensuring that the programming team is documenting overall standards, decisions, issues and project specific processes will make it easier for those programmers to get up to speed. Creating a cohesive core team of programmers who are working well together, producing code that is easily understood/adopted by others, sharing best practices, and documenting those practices means the team can be “built out” more easily over time.

The following questions can help you ensure you are building the team:

- Have you evaluated the various programming tasks, ensuring that “key” tasks can be easily supported by more than one programmer?
- Have you been able to cross-train programmers so they are learning from each other/sharing workload?
- Have you assigned interesting, meaningful work to each programmer that they can complete? Are they clear on what they need to do to be successful? Are they ready for a new challenge?
- Have you and the team established standard programming practices? Is the entire team using those practices/standards?
- How easily will it be to add new programmers to the team? Are there sufficient processes, documentation and mentors for them to come up to speed quickly?
DEVELOP EXTENSIVELY

Prioritizing “development” in your programming team is important in many ways; not only does it build up the skills of your team members to do their work better, faster, with higher quality, it also helps to retain that team. Programmers like to build reusable code to cut down on redundant tasks but also like to try new ways of doing things using the latest and greatest SAS® techniques. However, they need time to explore and apply those new methods. As a team leader, you can create forums where the team can share new techniques, review code, and/or discuss lessons learned to develop improved ways of programming. Additionally, time spent creating and improving a reusable, standard macro library that the whole team can utilize can be an exciting challenge for a team, led by one of your highly technical team members. Sharing reusable code and good programming practices can benefit both your senior and junior level programmers.

In our organization, we have had great success creating and maintaining forums for SAS® programmers to share tips, techniques, and practice presentation skills. Six years ago, we started a “Biogen Idec” SAS® Users Group (BIOSUG) which has continued to flourish. Six to eight times a year, we host a global forum for programmers and statisticians to share topics/issues/techniques/knowledge (4-5 presentations at each meeting). This year several team members also began a global “Programmer T time” to share very detailed code and offer suggestions to improve code. Lastly, programmers at Biogen Idec must present after attending any conferences in order to share that knowledge throughout the group. Through these forums, the team develops individually and as a global team.

The following questions can ensure you are prioritizing the development of team members:

- How will programmers share best practices? What’s the best forum/process?
- Do programmers have time each (week, month, etc.) to try new techniques and programming procedures?
- Are programmers expected to research external, new techniques? Will they be rewarded for doing so?
- How will programmers share improved techniques internally? Externally?
- Are there opportunities/development plans set up for both senior and junior level programmers?

RETAIN AND REWARD

Now that you’ve planned for a team, built the team, and have provided methods for development, there are still a few more things you can do to retain your team. Evaluate progress often and share how much is completed with the global team. Celebrate short term successes as well as the completion of the long term goal. Also, it may be beneficial to reward the team (or individuals on the team) for specific examples of excellence or achievement. However, be cautious – reward systems can backfire if not rolled out properly and in the right environment. Some team members abhor public recognition, some team members thrive on it. As you get to know your team members, you should provide appreciation and reward in the method that will be most well received.

Additionally, as the team becomes a highly functioning team, your role as a leader should become less directive and more supportive. Remember to communicate often but don’t get in their way. Help the team to proactively identify potential hurdles and overcome problems. Programmers like to work in an independent and flexible environment – check in on the health of the team to find out what is causing delays and/or what is needed to be even more productive. While you want the team to be challenged, you want to mitigate “overburdening” the team and may need to adjust workload or add team members in order to meet the goal.

The following questions can help you ensure you have ways to retain and reward the team:

- Are you evaluating progress and addressing delays/issues as well as celebrating progress?
- Are you providing reward/recognition for examples of excellence by individuals or the entire team?
- Are you providing reward/recognition in a manner that will be well received by the individuals and team?
- Are you leading the team in the appropriate manner at specific stages? Directing versus Supporting versus Collaborating?
- Are you periodically checking on the “health” of your team (not just progress of tasks but how the team is functioning)?
PROTECT THE TEAM

Lastly, teams are generally set up to achieve a series of tasks to reach an end goal and have a beginning and an end. In the midst of the project, we have to face the possibility that one or more of our team members may need to leave the team and that the team eventually should “dissolve” when the goal is completed. How can you anticipate or protect the team from disruption if a team member needs to leave the team and the team begins to naturally “shrink” when tasks are completed?

A lot of the tips we’ve already discussed certainly help to “protect the team” as these potential situations occur. By measuring interim progress of the work to know exactly where tasks stand, you will know what tasks are at risk. Sharing and splitting up the workload across programmers ensures that more than one person knows the details of a specific task and can pick up code easily. By checking on the “health” of your team, you will know how programmers individually are doing and if there is risk that they may need to leave to work on other projects or if they may even be at risk for leaving the company. Setting programming standards facilitates programmers easily understanding code that other programmers have completed and will protect you from that risk if a programmer may leave the team. Through extensive development of skills, “junior” level programmers can take on more “senior” level tasks and “senior” programmers can take on new challenges.

The following questions can ensure you can protect the team from disruption if one/more programmers leave the team:

- Periodically, check on the health of your team - Are any of your programmers “at risk”? If so, do you have a plan for someone else on the team to pick up that work? Would they be able to do that easily or are other actions required to mitigate that risk?
- Do you have assigned backups for key tasks and processes?
- Have you set up assignments with enough “cross-training” across the team to ensure that the team can be successful if one/more programmers need to leave the team?
- Do you have a plan/understanding of how the team will need to “shrink” as tasks are completed? As the goals are completed, you want to ensure that programmers are able to transition to other projects or tasks to keep doing that important, meaningful work.
- Lastly, as team members leave the team, have you provided a summary of their achievements and recognition of those achievements?

CONCLUSION

It is extremely rewarding to be part of a highly functioning programming team. In today’s environment, retaining and building a talented programming team can be a challenge. It takes a mixture of communication, interesting project work, development, leadership and support to keep programmers engaged and dedicated to the goal. Through these tips, we hope we have provided ideas that will help you keep your team “in for the long haul”.

ACKNOWLEDGMENTS

We would like to acknowledge the entire Biogen Idec Global Statistical Programming and Operations department as well as the cross-site Programming management team (Greg Silva, Tina Casteris, Paula Huhtakangas, Matthew Wien, Chris Kania, Eric Ding and Kathryn Spears).

I’d also like to acknowledge Greg Silva, John Hotaling, Vincent Daforo, Chris Kania, Tim Tian and Xiaodong Huang in their efforts to envision, create and maintain our long standing programming forums (Biogen Idec SAS® User’s Group) and Programmer “T” time to share best practices across programmers across sites.

Building and retaining such a talented team requires a lot of effort by both individual team members and their supervisors to work well in an ambiguous and at times stressful environment. We are able to do the important meaningful work for patients because of the strengths and efforts of this global team.
CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

  Name:  Lynn DiFinizio, Director, Statistical Programming and Operations
  Enterprise:  Biogen Idec
  Address:  14 Cambridge Center
  City, State ZIP:  Cambridge, MA 02142
  Work Phone: 617 679 3794
  Fax: 617 679 2526
  E-mail:  lynn.difinizio@biogenidec.com

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. © indicates USA registration.